CLAIMS:

1. A method of managing keys in a key distribution system for a communications group, the key distribution system maintaining a tree of nodes including at least one leaf node that has a parent node, each node of the group being associated with a first key,

the method comprising:

the system updating the first keys of a first branch of nodes in the tree by allocating new first keys to each of the nodes in the branch;

the system determining an offset for generating the updated first key of each node in the branch from the previous node in the branch; and

broadcasting each of said offsets so that, given the updated first key associated with the first node of said branch, each updated first key of said branch of nodes can be calculated.

2. A method as claimed in claim 1, wherein the first key of each parent node in said tree of nodes is generated from the first key of each of its child nodes by two one-way functions and a mixing function, the mixing function including the offset as a parameter.

3. A method as claimed in claim 2, wherein the mixing function in an XOR function.

4. A method as claimed in claim 2 or claim 3, wherein each parent key is generated using the formula f(f(child_key) XOR OFFSET), wherein OFFSET is the offset and f represents a one-way function and wherein child_key is the first key of a child node of said parent node.

5. A method as claimed in any preceding claim, wherein the communication group comprises at least one member that is associated with a leaf node of the tree of nodes.

6. A method as claimed in claim 5, wherein information transferred to, from or between members of the communication group is encrypted using an application data encryption key, the encryption key comprising a join field and a leave field, wherein each member of the group knows the join field of the encryption key, and wherein the leave field of the encryption key is derived from the first key of a root node of the tree.

7. A method as claimed in claim 6, wherein the join field of the encryption key is updated each time a member joins the group.

8. A method as claimed in claim 7, wherein the new member joins the group using the following method:

the new user requests access to the group;

the new user is granted access to the group;

the new member is assigned a node at a new leaf node of the communication group;

the new member is sent all the information required to generate the first key of each node on a branch of nodes from the new leaf node to the root node; and

the join field of the application data key is updated.

9. A method as claimed in claim 8 wherein the method further comprises:

the generation of a new node as the parent of both the new leaf node and a pre-existing node.

10. A method as claimed in any one of claims 7 to 9, wherein the updated join field is generated from the previous join field using a one-way function.

11. A method as claimed in any one of claims 6 to 10, wherein a key update request is generated each time a member leaves the group, wherein the first keys of each node of the branch of nodes including both the node associated with the member that is leaving the group and the root node are the keys that are updated.

12. A method as claimed in claim 11, wherein a member leaves the group using the following method:

an instruction to remove a member from the group is generated;

the parent node of the node associated with the leaving member is deleted;

the sibling node of the node associated with the leaving member is promoted to the position occupied by the deleted node;

the first key of each node on the branch of nodes from the promoted node to the root node is updated;

offset messages for generating the new first keys are broadcast to the group;

remaining members of the communications group calculate the updated first key nodes of the tree.

13. A method as claimed in claim 12, wherein the instruction to remove a member from the group is generated by the member that is leaving the group.

14. A method as claimed in claim 12, wherein the instruction to remove a member from the group is generated by a key distribution server.

15. A method as claimed in any preceding claim, wherein the nodes are arranged in a hierarchical tree.

16. A method as claimed in claim 15, wherein the nodes are arranged in a binary tree.

17. A method as claimed in any preceding claim further including:

retransmitting messages enabling users to update keys in case the users have not received those messages.

18. A method as claimed in claim 17 wherein the retransmitted messages are attached to application data packets.

19. A method as claimed in claim 17 or claim 18 wherein the retransmitted messages contain a sequence number indicative of the position in the sequence of key updates.

20. A method as claimed in claim 19 wherein the sequence number is cyclic.

21. A key distribution system which, in operation, performs the method of any preceding claim.

22. A key distribution system for a communications group, the key distribution system maintaining a tree of nodes including at least one leaf node that has a parent node, each node being associated with a first key, wherein:

the first key of each parent node in the tree is derived from the first key of each of its child node by two one-way functions and a mixing function, the mixing function including an offset value as a parameter.

23. A key distribution system as claimed in claim 22, wherein the mixing function in an XOR function.

24. A key distribution system as claimed in claim 22 or claim 23, wherein each parent key is generated using the formula f(f(child_key) XOR OFFSET), wherein OFFSET is the offset and f represents a one-way function and wherein child_key is the first key of a child node of said parent node.

25. A key distribution system as claimed in any one of claims 22 to 24, wherein:

the first keys of a first chain of nodes along a branch of the tree are updated by allocating new first keys to each of those nodes in response to a request to update the first keys of that chain of nodes;

an offset for generating the updated first key of each member of the chain from the previous member of the chain is determined; and

each of said offsets is broadcast so that, given the updated first key associated with the first node of said chain of nodes, each updated first key on said chain of nodes can be calculated.

26. A key distribution system as claimed in any one of claims 22 to 25, wherein the communication group comprises at least one member that is associated with a leaf node.

27. A key distribution system as claimed in any preceding claim, wherein information transmitted to, from or between members of the communication group is encrypted using an application data encryption key, the encryption key comprising a join field and a leave field, wherein each member of the group knows the join field of the encryption key, and wherein the leave field of the encryption key is derived from the first key of a root node of the tree.

28. A key distribution system as claimed in claim 27, wherein the join field of the encryption key is updated each time a member joins the group.

29. A key distribution system as claimed in claim 28, wherein the new member joins the group using the following method:

the new user requests access to the group;

the new user is granted access to the group;

the new member is assigned a node at a new leaf node of the communication group;

the new member is sent all the information required to generate the first key of each node on a branch of nodes from the new leaf node to the root node; and

the join field of the application data encryption key is updated.

30. A key distribution system as claimed in claim 29 wherein the said new member join method further comprises the generation of a new node as the parent of both the new leaf node and a pre-existing node.

31. A key distribution system as claimed in any one of claims 28 to 30, wherein the updated join field is generated from the previous join field using a one-way function.

32. A key distribution system as claimed in any one of claims 27 to 31, wherein a key update request is generated each time a member leaves the group, wherein the first keys of each node of the branch of nodes including both the node associated with the member that is leaving the group and the root node are the keys that are updated.

33. A key distribution system as claimed in claim 32, wherein a member leaves the group using the following protocol:

an instruction to remove a member from the group is generated;

the parent node of the node associated with the leaving member is deleted;

the sibling node of the node associated with the leaving member is promoted to the position occupied by the deleted node;

the first key of each node on the branch of nodes from the promoted node to the root node is updated;

offset messages for generating the new first keys are broadcast to the group;

remaining members of the communications group calculate the updated first keys of nodes of the tree.

34. A key distribution system as claimed in claim 33, wherein the instruction to remove a member from the group is generated by the member that is leaving the group.

35. A key distribution system as claimed in claim 33, except, wherein the instruction to remove a member from the group is generated by the key distribution server.

36. A key distribution system as claimed in any one of claims 22 to 35, wherein the nodes are arranged in a hierarchical tree.

37. A key distribution system as claimed in claim 36, wherein the nodes are arranged in a binary tree.

38. A key distribution system for a communications group, the key distribution system comprising an encryption key and maintaining a tree of nodes including a root node that has at least one child node, and at least one leaf node that has a parent node, the communication group comprising at least one member, wherein the encryption key comprises a join field and a leave field, and wherein:

each member of the group knows the join field of the encryption key;

each node of the key distribution system is associated with a leave key;

the leave field of the encryption key is derived from the leave key of the root node.

39. A key distribution system as claimed in claim 38, wherein said at least one member is associated with a leaf node of the tree of nodes.

40. An encryption key as claimed in claim 38 or claim 39, wherein the join field of the encryption key is updated each time a member joins the group.

41. An encryption key as claimed in claim 40, wherein the updated join field is generated from the previous join field using a one-way function.

42. An encryption key as claimed in any one of claims 39 to 41, wherein a key update request is generated each time a member leaves the group, wherein the leave keys of each node of the branch of nodes including both the node associated with the member that is leaving the group and the root node are updated.

43. An encryption key as claimed in any one of claims 39 to 42, wherein the first key of parent nodes in the tree is generated from the first key of each of its child nodes by two one-way functions and a mixing function, the mixing function including an offset as a parameter.

44. An encryption key as claimed in claim 43, wherein the mixing function in an XOR function.

45. An encryption key as claimed in claim 43 or claim 44, wherein each parent key is generated using the formula f(f(child_key) XOR OFFSET), wherein OFFSET is the offset and f represents a one-way function and wherein child_key is the first key of a child node of said parent node.